

JasonLee的专栏

一个写着几行代码的理工科文学青年。

目录视图

摘要视图

RSS 订阅

个人资料



jasonblog

访问：1185789次

积分：12857

等级：BLOG

排名：第638名

原创：192篇

转载：9篇

译文：2篇

评论：845条

文章搜索

博客专栏



CPython学研笔记

文章：14篇

阅读：25890



Inside The iOS Dev

文章：16篇

阅读：221104

阅读排行

iOS中block实现的探究

(80046)

再谈iOS 7的手势滑动返回

(53411)

如何更好地限制一个UITextView

(52609)

在iOS上实现一个简单的聊天应用

(38815)

来淘宝的这一年：前篇、后篇

(34773)

iOS应用的crash日志的分析

(31002)

随手记Swift基础和Optional Type

(24552)

使用UIImageView展现网络图片

(21273)

【CSDN会员专属福利】OpenStack Days China 大会门票，先到先得

【收藏】Html5 精品资源汇集

我们为什么选择Java

随手记Swift基础和Optional Type（问号?和感叹号!）

2014-06-06 15:07

24556人阅读

评论(2)

收藏

举报

分类：iOS (85)

业界 (10)

版权声明：本文为博主原创文章，未经博主允许不得转载。

距离Apple推出Swift已经有几天了，网上也时不时出现“急招Swift程序员，要求有一天工作经验”的帖子。看到Swift，除了苹果放的另外一门语言的链接（http://swift-lang.org/），还可能联想到Taylor Swift，或者铃木雨燕。

这几天网上关于Swift的话题和材料应该迅速膨胀，比如StackOverflow的相关标签，还有知乎上的一些相关讨论。

关于Swift的推出，我个人觉得对现有的使用Objective-C的iOS开发者不算是很大的冲击，毕竟写代码这门功夫，比较注重内功修炼。而对于功能越强大越灵活的语言，有可能高手拉开的距离更大，比如我很久以前转载的一篇如何消除Python列表中的重复元素。

初步看起来，觉得Swift有点像JavaScript，又有点Python的风格，再看还有C++的特性。总的下来，有点“杂糅”或“博采众长”的味道，以下是一些基础代码示例：



为了体现playground的效果，我在这里上传了截图，而不是直接贴代码。说到这个，可见从某种角度来看，大家都在同一起跑线上，毕竟连怎么看日志输出都可以在SO上收获很多reputation，哈哈。

上面的代码我只截取了一部分，还有部分如下：

```
[cpp]
1. var arr = [0, 1, 2]
2. for item in arr {
3.     println(item)
4. }
5.
6. var dict = ["k1": "v1", "k2": "v2"]
7. for item in dict.keys {
8.     var s = item + dict[item]!
9.     println(s)
10. }
```

http://blog.csdn.net/jasonblog/article/details/28876823

1/4

Core Data浅谈系列之三

(19371)

iOS中注册功能的体验探!

(18016)

价值链接

刘未鹏 | MIND HACKS

酷壳 – CoolShell.cn

阮一峰的网络日志

弯曲评论

DBA Notes

云风的 BLOG

Erlang非业余研究

追风刀-丁奇

Joshua Zhu's Blog

Tim[后端技术]

友情博客

Judy的心情杂货铺

Nancy's Blogspot

SmoClock

KISimple 的专栏

code6 - wzc

文章分类

iOS (86)

Core Data (15)

Python (21)

C/C++ (27)

InSearchOfCS (8)

Linux (15)

Mac (2)

Qt (9)

WebDev (4)

Others About CS (15)

业界 (11)

职业 | 工作 (36)

生活 | 旅行 | 随笔 (32)

最新评论

在iOS App中增加完整的照片多选MoreFocus: 这个是我见过最好的多选框架  
https://github.com/Jameson-zxm/MSImag...

iOS中block实现的探究  
kangqijun: 写的很好, 但是有一点想不通的地方, 想请您解惑一下: "当block从栈上被copy到堆上时, 会调用\_\_..."

iOS中block实现的探究  
S往事随风S520: 谢谢分享, 看了一遍 云里雾里的, CPP里的内容感觉像是在看天书, 收藏了 慢慢研究

iOS中线程Call Stack的捕获和解: qweq123123123123: 有demo吗, 你的代码没有给全哈, 跑不通。

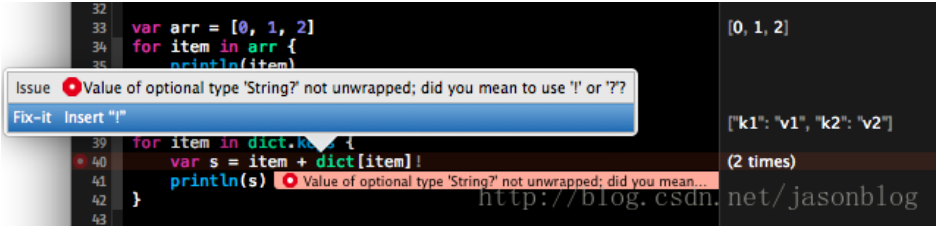
从NSTimer的失效性谈起 (二): jasonblog: @wwwang89123: 可以参考下这里  
http://stackoverflow.com/q/26...

从NSTimer的失效性谈起 (二): wwwang89123: 楼主, 自带的NSTimer, 退到后台是可以继续执行的, 而GCD 定义的Timer, APP退到后台就暂...

从NSTimer的失效性谈起 (二):

我随手写了遍历数组和字典的代码, 结果遇到问题了。

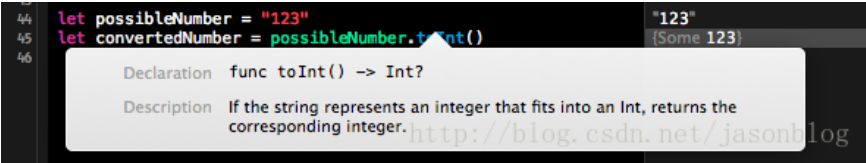
注意倒数第三行末尾有个感叹号! 如果不加这个感叹号, 会报错:



这里提到了“optional type”、“String?”和“not unwrapped”等概念。

在《The Swift Programming Language》的Language Guide - The Basics这一章中, 紧跟Tuples小节的就是Optionals小节 (之前的A Swift Tour也有掠过)。

根据这一小节开头的两行示例代码, 我们可以得知有一种新类型是在既有类型后面加问号? :



这种类型叫做“Optional Type”。参考Swift的类型声明, 我们尝试声明Optional类型看看:

```
[cpp] C ?
01. var oVar: Optional
```

编译器会报错:

```
[cpp] C ?
01. Playground execution failed: error: <REPL>;46:11: error: reference to generic type 'Optional' req
02. var oVar: Optional
03. ^
04. generic type 'Optional' declared here
```

根据错误提示, 我们可以得知这是个泛型类型, 需要指定具体的类型:

```
[cpp] C ?
01. var oVar: Optional<Int>
```

于是我们可以跳进去查看Optional的具体定义:

```
[cpp] C ?
01. enum Optional<T> : LogicValue, Reflectable {
02.     case None
03.     case Some(T)
04.     init()
05.     init(_ some: T)
06.
07.     /// Allow use in a Boolean context.
08.     func getLogicValue() -> Bool
09.
10.     /// Haskell's fmap, which was mis-named
11.     func map<U>(f: (T) -> U) -> U?
12.     func getMirror() -> Mirror
13. }
```

此时再看剩下的代码:

wwwang89123: @L\_H\_Y\_1990: 确实，这个源码看了也忘记了，不大好理解了

从NSTimer的失效性谈起（二）：lvable: 前辈感觉你分析的所有东东，都是通过源代码分析的。如果写一篇博客关于分析技术心得和方法就最好啦，哈哈，...

iOS中block实现的探究lvable: 写的好强悍，绝对是我这个菜鸟的礼物

关于AssetsLibrary无法获取本地相册dongtinghong: 好像ios8以上的系统要用PHAsset才能检索出所有相册，而且检索出的相册组归类有点问题，楼主有没...

文章存档

- 2016年04月 (1)
- 2016年03月 (2)
- 2015年11月 (2)
- 2015年05月 (1)
- 2014年12月 (1)

展开



再结合官方文档的描述，可以得到如下几点结论：

1. Int? 等价于 Optional<Int>，是一种语法糖；
2. 上述 Int? 类型的变量 oVar 表示 oVar 要嘛包含Int类型的数据，要嘛为nil —— 非Optional Type的变量不能赋值为nil；也就是说，Optional类型允许变量没有值，其它类型如果没有初始化值在使用时会报错 —— 比如通过下标访问数组元素时返回Optional类型，越界时可以返回nil；
3. 在声明一个Optional类型变量时如果没有赋初始值，那么默认为nil；
4. Optional类型变量的真实值是被封装起来的，包装在其枚举值Some中，如上{Some 123}；
5. 使用问号？表示封装（可能有值可能没有），使用感叹号！表示拆封取值（强制认为有值，如果没有则会触发运行时错误），有点类似“Has a value?” —— "I assure it has!"；
6. 由于Optional枚举类型遵循LogicValue协议，所以可以作为逻辑判断条件，当有值时为true，不包含值时为false；

在ARC这一章中有一节叫“Unowned References and Implicitly Unwrapped”讨论了一种适用场景，除此之外，Optional Chaining也给了整整一个章节，这里就不继续展开了。

：



- 上一篇 随手记UIKit Dynamics
- 下一篇 Core Data浅谈系列之十： 关于数据模型中实体的属性

我的同类文章

iOS (85)	业界 (10)
<ul style="list-style-type: none"><li>记一次帮同事排查由命名引发.. 2016-04-20 阅读 1943</li><li>从NSTimer的失效性谈起（一.. 2016-03-15 阅读 995</li><li>iOS中线程Call Stack的捕获... 2015-11-18 阅读 1839</li><li>在iOS中读取通讯录中联系人... 2014-12-03 阅读 3230</li><li>利用objc的runtime来定位次... 2014-06-17 阅读 5349</li></ul>	<ul style="list-style-type: none"><li>从NSTimer的失效性谈起（二... 2016-03-15 阅读 1180</li><li>iOS中线程Call Stack的捕获... 2015-11-18 阅读 1847</li><li>Auto Layout和UILabel 2015-05-25 阅读 3129</li><li>再谈multistage text input（中... 2014-07-04 阅读 8912</li><li>Core Data浅谈系列之十： ... 2014-06-08 阅读 4793</li></ul>
<a href="#">更多文章</a>	

猜你在找

- Swift中的问号和感叹号
- Swift语言中问号 和 感叹号 的作用
- Swift中的问号和感叹号
- Swift语言中问号 和 感叹号 的作用
- 浅谈Swift20对optional value的安全处理慎用问号叹



支架泳池



app开发报价



留学生公寓



编写



苹果笔记本电



app设计



app外

查看评论

2楼

OneByte\_Code

2014-08-08 12:06发表



江哥?

1楼

Macist

2014-07-31 17:02发表



被楼主清晰思路所折服。  
1.这是个泛型类型，需要指定具体的类型  
2.“Has a value?” —— “I assure it has!”

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

apttech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap